بسمه تعالی

وزارت ارتباطات و فناوری اطلاعات

سازمان فناوری اطلاعات ایران

معاونت امنیت فضای تولید و تبادل اطلاعات

گزارش آسیب‌پذیری Check Point Gaia Portal

گزارش فنی

نوع سند ..... گزارش فنی

شماره نگارش ۱

تاریخ نگارش ۱۴۰۲/۰۵/۱۸

طبقه‌بندی سند ..... **عادی**

تهران، خیابان شهید بهشتی بین بزرگراه شهید مدرس و خیابان احمد قصیر- پلاک۲۶۷

(۰۲۱) ۸۸۱۱۵۷۲۴ (۰۲۱) ۸۸۱۱۵۷۲۴ cert.ir

فهرست مطالب

۱ شرح آسیب‌پذیری

محققان Pentests یک آسیب‌پذیری Command Injection با شناسه CVE-2023-28130 در پورتال Gaia محصول Check Point را کشف و گزارش نموده‌اند. این آسیب‌پذیری به یک حمله‌کننده با احراز هویت و دسترسی نوشتن در تنظیمات DNS ، اجازه می‌دهد تا از طریق تزریق دستور، کد مخرب را از راه دور اجرا نماید. شدت این آسیب‌پذیری با امتیاز ۸,۴ و شدت بالا گزارش شده است.

Gaia نامی که از اساطیر یونانی اقتباس شده است، نام سیستم عاملی است که برای برنامه‌های امنیتی توسط Check Point استفاده می‌شود. به طور خلاصه، Gaia یک سیستم عامل امنیتی یکپارچه است که سیستم‌عامل‌های اصلی Check Point مانند IPSO و سیستم‌عامل دستگاه‌های امنیتی را با یکدیگر ترکیب می‌کند.

این سیستم از یک واسط وب به نام Gaia Portal برای پیکربندی استفاده می‌کند. این پورتال تقریباً تمام تنظیمات سیستم را از طریق وب ارائه می‌دهد و آسیب‌پذیری مورد بررسی در همین پورتال کشف شده است. در این آسیب‌پذیری، حمله‌کننده با تزریق دستورات، می‌تواند کد مخرب را در یک اسکریپت CGI اجرا کرده و به صورت از راه دور دسترسی به سیستم را کسب کند. این موضوع به طور قابل توجهی امنیت سیستم را تهدید می‌کند.

آسیب‌پذیری در پارامتر "hostname" و در اسکریپت "cgi-bin/hosts_dns.tcl" شناسایی شده است. در این مورد، اگر یک کاربر دارای جلسه (Session) معتبر باشد و دسترسی نوشتن در تنظیمات DNS را داشته باشد، این کاربر می‌تواند دستورات خود را اجرا کند. این دستورات با امتیاز "Admin" اجرا می‌شوند.

آسیب‌پذیری در حین یک فرآیند تست نفوذ وب شناسایی شده است. محققان اعلام کرده‌اند که در طول این فرآیند، هر نوع ورودی به عنوان یک آسیب‌پذیری بالقوه در نظر گرفته شده است. ورودی‌ها اگر به درستی استفاده نشوند، می‌توانند منجر به انواع آسیب‌پذیری‌های تزریقی مانند تزریق SQL ، XSS، تزریق قالب و تزریق دستور شوند. در این موارد، ما با آسیب‌پذیری Command Injection مواجه شده‌ایم.

محققان با افزودن مقدار "|command here" به انتهای پارامتر "hostname" به این آسیب‌پذیری دست پیدا کردند. این آسیب‌پذیری بدون نیاز به بررسی کد و در قالب یک روش "grey box" کشف شده است. این به این معناست که محققان اطلاعات کد منبع برنامه را مورد بررسی قرار نداده‌اند ولی با آزمون‌های نفوذی خاص، این آسیب‌پذیری را کشف کرده‌اند.

اگه کد اسکریپت زیر را بررسی کنیم :

```
/!#usr/bin/ipstcl2
source debug.tcl
variable dFile "/tmp/hosts_dns.post.debug"
namespace eval ::hosts}
      if {[catch {libdb init -local} myDb} {[
#             failed to initialize!!!
            lappend err_list "Unable to connect to database ${myDb"{
            exit
{
      ipso -nohtml
      set method [get_Method[
      MIME text
      printDebug
#       URL argument to indicate which data is needed
      set option [getVal option[
      global Q_Names
      if {$method == "POST} {"
            source validate.tcl
            source subs.tcl
            source showResult.tcl
            set result 0
            set error_flag 0
            set err_list [list[
            set set_list [list[

            set result [Validate $Q_Data err_list[
            if {$result == 0} {
                  foreach var $Q_Names}

                        if {$var == "domainname} {"
                              lappend set_list domainname [getVal $var[
 {                        elseif {$var == "suffix} {"
                              lappend set_list resolv:domain:1 [getVal $var[
 {
[...]
 {                        elseif {$var == "hostname} {"
                              lappend set_list machine:hostname [getVal $var[
{
[...]

                        if {$var == "save} {"
                              lappend set_list :save""
{
```

```
{
{

        dmsg "set_list: $set_list"
        if {[llength $set_list] > 0} {
            set cmd [concat [list libdb set $myDb -list ] $set_list[
            set result [catch $cmd err_list[
{

        dmsg "Result: $result"
        dmsg "err_list: $err_list"
        set buf [::showResult::generateJson $result [getVal save] $err_list[

        HTML $buf
        dmsg $buf
{       else}
#GET
        source json.tcl
```

وقتی درخواست از نوع POST باشد، تمام متغیرها با استفاده از یک ساختار if-elseif بررسی می‌شوند. در اینجا سه پارامتر اصلی به نام‌های"hostname" ، "domainname"و "save" ارسال می‌شوند که با استفاده از شرط‌های if-elseif بررسی می‌شوند.

متغیر نهایی مجموعه "set_list" به صورت زیر تشکیل می‌شود:

۱. اگر پارامتر "hostname" با مقدار "hostname_value" ارسال شود :مقدار "machine: hostname <hostname_value>" به "set_list" اضافه می‌شود.

۲. اگر پارامتر "domainname" با مقدار "domainname_value" ارسال شود :مقدار "domainname <domainname_value>"به "set_list" اضافه می‌شود.

۳. اگر پارامتر "save" با مقدار "true" تنظیم شود :مقدار ":save" به "set_list" اضافه می‌شود.

به این ترتیب، متغیر نهایی "set_list" می‌تواند از ترکیب این مقادیر حاصل شود.. مقدار نهایی لیست "set_list"به شکل زیر است:

```
machine:hostname <hostname_value> domainname <domainname_value> :save
```

در ادامه کد بالا ، به خطی که دستور زیر اجرا میشود می‌رسیم:

```
set cmd [concat [list libdb set $myDb -list ] $set_list]
```

در این خط، cmd مقدار زیر را میگیرد :

```
libdb set db_0 -list machine:hostname <hostname_value> domainname <domainname_value> :save.
```

در خط بعدی، دستور catch با مقدار لیست cmd فراخوانی می‌شود. دستور catch یک برنامه یا فرآیند را اجرا کرده و کد خروجی را برگردانده می‌کند.

۲ مراجع

https://pentests.nl/pentest-blog/cve-2023-28130-command-injection-in-check-point-gaia-portal/